providing a physical symbol-bearing reel including an encoder for indicating a position of said reel;

providing a reel controller for performing low-level operations related to movement of said reel;

providing a central processing unit for issuing high-level commands to said reel controller related to the movement of said reel;

sending a command from said central processing unit to said reel controller to determine a type of said encoder; and

determining the type of said encoder with said reel controller, which includes causing a motor to spin said reel and detecting a physical characteristic of said encoder.

REMARKS

Claims 1-5, 7-23 and 27-29 remain in the application for prosecution. Claims 6 and 24-26 have been cancelled. Claim 23 has been amended to include the limitations of cancelled claims 25 and 26. Claim 27 has been amended to depend from amended claim 23, instead of cancelled claim 26. Claim 28 has been rewritten in independent form to include the limitations of base claim 23 (prior to amendment) and intervening claim 25. No new issues should be presented by the amendments. Submitted herewith is a clean set of pending claims.

Claim Rejections - 35 U.S.C. § 103

a. Claims 1-5 and 7-28

Claims 1-5 and 7-28 were rejected under 35 U.S.C. § 103(a) as obvious over U.S. Patent No. 6,394,900 to McGlone et al. ("McGlone"). The Applicant respectfully disagrees for the reasons set forth below. As noted above, claims 24-26 have been cancelled. Claim 23 has been amended to include the limitations of cancelled claims 25 and 26. Claim 27 has been amended to depend from amended claim 23, instead of cancelled claim 26. Claim 28 has been rewritten in independent form to include the limitations of base claim 23 (prior to amendment) and intervening claim 25.

Claims 1-5, 7-23 and 27 require that *the central processing unit (CPU) sends configuration data to the local microcontroller for configuring the local microcontroller to a reel spinning game conducted with the slot machine.*

The Examiner argues in the alternative. On the one hand, the Examiner asserts:

> McGlone also teaches that the master gaming controller includes a memory storing software for device drivers for at least some of the slot reel peripherals. (Col 3, 35-41) These device drivers are configuration data and are sent to the local microprocessor for configuring it to a reel spinning game conducted with a slot machine.

In response to this contention, the Applicant strongly disagrees. *Software device drivers cannot be equated with configuration data.* A device driver is a program that controls a device, such as a printer, disk drive, or keyboard. Many drivers, such as the keyboard driver, come with an operating system. For other devices, you may need to load a new driver when you connect the device to a computer. A driver acts like a translator between the device and programs that use the device. Each device has its own set of specialized commands that only its driver knows. Most programs, however, access devices by using generic commands. The driver, therefore, accepts generic commands from a program and then translates them into specialized commands for the device. *See* www.webopedia.com/TERM/d/driver.html. In sharp contrast, *configuration data are distinct pieces of information and are different from programs such as device drivers.* Programs are collections of instructions for manipulating data. *See* www.webopedia.com/TERM/d/data.html. Examples of configuration data include the type of slot machine, number of symbols on a reel, number of steps per revolution of a stepper motor, number of pulses per step, etc.

On the other hand, the Examiner alternatively asserts:

> McGlone teaches that, "The peripheral controller may have a non-volatile memory arranged to store configuration parameters specific to the slot reel peripheral and state history information of the slot reel peripheral. In one embodiment, the non-volatile memory might be used to store the configuration parameters needed to drive the slot reel using the drive mechanism including a moment of inertia of the slot reel, the size of the slot reel and one or more acceleration parameters." (Col 3, 17-24) The suggestion that the peripheral (i.e., local) controller may have non-volatile memory for storing configuration data provides a strong suggestion that the opposite may also be true. If the peripheral controller does not have non-volatile memory for storing configuration data, that data must be loaded to the peripheral controller from the central processing unit. Doing this would eliminate unnecessary duplication of parts because the data could be stored in one set of non-volatile memory instead of on non-volatile memory associated with each peripheral controller. This would reduce the cost of the gaming machine.

Again, the Applicant strongly disagrees. First, if McGlone were modified to load configuration data to the peripheral controller from the CPU instead of from a non-volatile memory in the peripheral controller, one would not eliminate unnecessary duplication of parts. The peripheral controller would still require volatile or non-volatile memory for receiving the configuration data from the CPU. The peripheral controller would also still require non-volatile memory for such other items as the software program containing the low-level instructions executed by the peripheral controller. Second, even if the peripheral controller required less non-volatile memory, any cost savings would be quite negligible. Accordingly, it would not have been obvious to modify McGlone to have its CPU send configuration data to the local microcontroller. Any conclusion to the contrary is based on hindsight using the Applicant's own teachings. It is the Applicant, not McGlone, that teaches that configuration data may be loaded to the reel controller from the CPU. The Examiner's asserted motivations, i.e., eliminating unnecessary duplication of parts and reducing costs, for modifying McGlone do not exist.

Claims 15, 17, 19 and 22 further require that *in response to the configuration data, the local microcontroller processes the configuration data and reports a status of configuration of the local microcontroller back to the CPU.* The Examiner contends that "[i]t would have been obvious . . . [for McGlone] to have communicated the status of configuration from the local processor to the CPU . . . in order to prevent the game machine from operating in a misconfigured condition that would lead to errors in the game."

The above contention incorrectly assumes that McGlone suggests transmission of configuration data by the CPU to the local microcontroller and processing of that data by the local microcontroller. As noted above, McGlone provides no such suggestion. Furthermore, it is quite common for configured peripherals in gaming machines to not report any status back to the CPU because the possibility of error is quite remote and because such a status report would complicate the CPU's software and lengthen the time required to boot up the gaming machine. Therefore, significant motivations exist for not reporting a status of configuration of the local microcontroller back to the CPU. Accordingly, it would not have been obvious to modify McGlone to report a status of configuration of its peripheral controller back to the CPU.

Claims 23 and 27 further require determining the type of an encoder (used for indicating reel position) with the reel controller, sending configuration data from the CPU to the reel controller, *and using the reel controller to compare the determined type of the encoder with the configuration data*. The Examiner applies the same rationale as applied to claims 15, 17, 19 and 22 to allege that this claimed feature would have been obvious -- namely, that it would have been obvious in order to prevent the gaming machine from operating in a misconfigured condition that would lead to errors in the game. As stated above, however, it is quite common for configured peripherals in gaming machines to not validate configurations because the possibility of error is quite remote and because validation would complicate the CPU's software and lengthen the time required to boot up the gaming machine. Contrary to the Examiner, there is no suggestion or motivation to modify McGlone.

Independent claim 28 requires *determining the type of the encoder with the reel controller by causing a motor to spin the reel and detecting a physical characteristic of the encoder*. The Examiner asserts:

> [t]he easiest way to determine the type of encoder present would be to cause the motor to spin and count the number of flags on the encoder. Thus it would have been obvious . . . to have caused the motor to spin the reel and detect the physical characteristic (i.e., the number of flags) of the encoder in order to determine which type of encoder was present, thus enabling the loading of the correct device driver.

McGlone, however, explicitly teaches that its master gaming controller would be able to identify the type of peripheral device and its features from a series of numbers. Column 18, lines 5-7. As an example, McGlone teaches that combinations of the device class, manufacturer, device protocol and serial number information from a particular device may be used. Column 18, lines 2-4. McGlone says nothing whatsoever about determining the type of encoder with the reel controller by causing a motor to spin the reel and detecting a physical characteristic of the encoder. Clearly, McGlone must not believe that this is the "easiest way" to determine the type of encoder. Assuming *arguendo* that the "easiest way" to determine the type of encoder is by causing a motor to spin the reel and detecting a physical characteristic of the encoder, the mere fact that McGlone clearly teaches a different way would lead an ordinary artisan away from the claimed invention. *See* MPEP 2141.02 (prior art must be considered in its entirety, including disclosures that teach away from the claims). To modify McGlone would change its principle of operation. *See* MPEP 2143.01 (the proposed modification cannot change the principle of operation of the reference).

b. Claim 6

Claim 6 was rejected under 35 U.S.C. § 103(a) as obvious over McGlone in view of U.S. Patent No. 5,102,136 to Heidel et al. Claim 6 has been cancelled so this rejection is moot.

c. Claim 29

Claim 29 was rejected under 35 U.S.C. § 103(a) as being unpatentable over McGlone in view of U.S. Patent No. 6,315,663 to Sakamoto ("Sakamoto"). Claim 29 requires that *the CPU issues a high-level command to the reel controller for informing the reel controller of an acceleration or deceleration profile.* This feature is advantageous because it reduces the processing burden of the CPU and frees up the CPU for other tasks. The Examiner acknowledges that McGlone "does not teach an acceleration or deceleration profile for accelerating or deceleration the reel." Instead, the Examiner relies upon Sakamoto for this teaching, saying that "[i]t would have been obvious . . . to have the CPU send high-level commands concerning acceleration or deceleration profile for accelerating or decelerating the reel to the reel controller in order to add visual interest to the slot machine game."

The Applicant respectfully submits that the combination of McGlone and Sakamoto is improper because McGlone clearly teaches away from the combination. Specifically, McGlone discloses that a table of motor step rates is stored in a memory of either the master gaming controller or the slot reel controller, depending upon which controller controls the reel's stepper motor. *See* column 9, lines 1-11; column 9, lines 23-57. *When the slot reel controller is used,* *McGlone's table of step rates preferably resides in the slot reel controller and is not transferred from the master gaming controller to the slot reel controller*:

> . . . Further, the maintenance required to replace a slot reel might be reduced when a slot reel peripheral is used. For example, the table of step rates is usually required to drive the stepper motor for a particular slot reel. When a new slot reel is installed on a gaming machine and the master gaming controller drives the stepper motor of the slot reel, a table to drive the stepper motor corresponding to the new slot reel may be loaded into a memory device on the mother board containing the master gaming controller. Further, all the low-level commands and software needed to drive the stepper motor may also be loaded in memory on the motherboard. This process may be very time consuming. Using the slot reel peripheral, most of this information may be contained within the slot reel controller. Column 10, lines 4-22.

In an example, McGlone teaches that the step rates are included in a low-level command issued by the slot reel controller, not a high-level command sent from the master gaming controller to the slot reel controller:

> . . . For the stepper motor example described above, the low-level commands, charge the motor, initiate first step, step at rate 1, step at rate 2, step at rate 3, step at rate 4, perform final step, and stop the motor, might be initiated by the slot reel controller 402 after receiving a high-level instruction from the master gaming controller 422 like "move the slot reel 420 to position A." Column 9, lines 51-57.

To modify McGlone based on Sakamoto would change McGlone's principle of operation. *See* MPEP 2143.01 (the proposed modification cannot change the principle of operation of the reference). Accordingly, the rejection of claim 29 should be withdrawn.

Conclusion

It is the Applicant's belief that all of the claims are now in condition for allowance and action towards that effect is respectfully requested.

If there are any matters which may be resolved or clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at the number indicated.

Respectfully submitted,

Date: February 19, 2003

Michael Blankstein

Michael J. Blankstein
Reg. No. 37,097
Attorney for Applicant